



received user input thereby producing a result that identifies incompatibilities between the domain members caused by the at least one selected domain member; and

a bounceback detection module operatively coupled to the configuration engine, the bounceback detection module adapted to modify the result by detecting and eliminating incompatibilities caused solely by bounceback behavior.

6. The system of claim 5, further comprising:

a page generation module operatively coupled to the configuration engine, the page generation module adapted to generate a configuration page based on the modified result so that domain members identified as being incompatible due to bounceback behavior are not marked as conflicted choices on the configuration page, and to provide the configuration page to the user.

7. The system of claim 5, wherein the configuration engine and the bounceback detection module are implemented by a set of software instructions running on a computer.

8. A method for performing a product configuration, the product configuration associated with a configuration problem defining a number of constraints, one or more variables, and domain members associated with each variable, the method comprising:

receiving user input specifying at least one selected domain member;  
 propagating the constraints over the received user input thereby producing a result that identifies incompatibilities between the domain members caused by the at least one selected domain member;  
 modifying the result by detecting and eliminating incompatibilities caused solely by bounceback behavior;

generating a configuration page based on the modified result so that domain members identified as being incompatible due to bounceback behavior are not marked as conflicted choices on the configuration page; providing the configuration page to the user; and repeating the receiving, propagating, modifying, generating, and providing steps until the product configuration is complete.

9. A method for detecting bounceback behavior associated with a configuration problem, the configuration problem defining a number of constraints, one or more variables, and domain members associated with each variable, the method comprising:

receiving a domain member selection for a particular variable; setting a bounceback detection bit vector associated with each non-selected domain member of the particular variable so that each of those bounceback detection bit vectors indicates bounceback behavior; setting an elimination flag associated with each non-selected domain member of the particular variable so that each of those elimination flags indicates that its associated domain member is tentatively eliminated; propagating the constraints to identify eliminated domain members of the variables; setting the bounceback detection bit vector of the eliminated domain members to indicate which variable caused their elimination; and setting the elimination flag of each of the other eliminated domain members.

10. The method of claim 9, further comprising preliminary steps of: initializing the bounceback detection bit vector for each domain member of each variable; and initializing the elimination flag for each domain member of each variable.

11. The method of claim 9, wherein the receiving step includes receiving a plurality of domain member selections associated with a corresponding number of

3 particular variables, and the setting and propagation steps of the method are performed  
4 for each of the domain member selections.

1 12. The method of claim 9, wherein bounceback detection bit vectors that  
2 indicate bounceback behavior indicate that the particular variable associated with the  
3 selected domain member is responsible for elimination of the non-selected domain  
4 members.

1 13. The method of claim 9, further comprising:  
2 confirming the tentative elimination of a non-selected domain member in  
3 response to the bounceback detection bit vector associated with that non-  
4 selected domain member not indicating bounceback behavior as a result of  
5 subsequent constraint propagation.

1 14. The method of claim 9, further comprising:  
2 overriding the tentative elimination of a non-selected domain member in response  
3 to the bounceback detection bit vector associated with that non-selected  
4 domain member indicating bounceback behavior despite subsequent  
5 constraint propagation.

1 15. The method of claim 9, wherein the step of setting the bounceback  
2 detection bit vector of an eliminated domain member to indicate which variable caused  
3 that domain member's elimination includes:  
4 based on the constraints, identifying a domain member causing the eliminated  
5 domain member to be eliminated; and  
6 copying the bounceback detection bit vector associated with the identified domain  
7 member to the bounceback detection bit vector associated with the  
8 eliminated domain member.

1 16. The method of claim 9, wherein the step of setting the bounceback  
2 detection bit vector of an eliminated domain member to indicate which variable caused  
3 that domain member's elimination includes:

4 based on the constraints, identifying a join corresponding to a disjunction;  
 5 logically ANDing the bounceback detection bit vectors associated with the  
 6 domain members included in the join thereby producing a resulting  
 7 bounceback detection bit vector; and  
 8 copying the resulting bounceback detection bit vector to the bounceback detection  
 9 bit vector associated with the eliminated domain member.

1 17. The method of claim 9, wherein the step of setting the bounceback  
 2 detection bit vector of an eliminated domain member to indicate which variable caused  
 3 that domain member's elimination includes:

4 based on the constraints, identifying a join corresponding to a conjunction;  
 5 logically ORing the bounceback detection bit vectors associated with the domain  
 6 members included in the join thereby producing a resulting bounceback  
 7 detection bit vector; and  
 8 copying the resulting bounceback detection bit vector to the bounceback detection  
 9 bit vector associated with the eliminated domain member.

1 18. The method of claim 9, further comprising:  
 2 generating a configuration page based on the constraints so that domain members  
 3 identified as being eliminated due to bounceback behavior are not marked  
 4 as conflicted choices on the configuration page; and  
 5 providing the configuration page to a user.

1 19. The method of claim 9, wherein the steps of the method are repeated each  
 2 time a user submits one or more new domain member selections.

1 20. The method of claim 9, wherein the method is implemented by a set of  
 2 software instructions running on a computer

1 21. A method for detecting and eliminating bounceback behavior associated  
 2 with a configuration problem, the configuration problem defining a number of

constraints, one or more variables, and domain members associated with each variable,  
the method comprising:

initializing a bounceback detection bit vector for each domain member of each  
variable;  
initializing a elimination flag for each domain member of each variable;  
receiving a domain member selection for a particular variable;  
setting the bounceback detection bit vector associated with each non-selected  
domain member of the particular variable so that each of those  
bounceback detection bit vectors indicates bounceback behavior;  
setting the elimination flag associated with each non-selected domain member of  
the particular variable so that each of those elimination flags indicates that  
its associated domain member is tentatively eliminated;  
propagating the constraints to identify eliminated domain members of the  
variables;  
setting the bounceback detection bit vector of the eliminated domain members to  
indicate which variable caused their elimination; and  
setting the elimination flag of each of the other eliminated domain members.

22. The method of claim 21, wherein the step of setting the bounceback  
detection bit vector of an eliminated domain member to indicate which variable caused  
that domain member's elimination includes:

based on the constraints, identifying a domain member causing the eliminated  
domain member to be eliminated; and  
copying the bounceback detection bit vector associated with the identified domain  
member to the bounceback detection bit vector associated with the  
eliminated domain member.

23. The method of claim 21, wherein the step of setting the bounceback  
detection bit vector of an eliminated domain member to indicate which variable caused  
that domain member's elimination includes:

0980429-032850

4 based on the constraints, identifying a join corresponding to a disjunction;  
5 logically ANDing the bounceback detection bit vectors associated with the  
6 domain members included in the join thereby producing a resulting  
7 bounceback detection bit vector; and  
8 copying the resulting bounceback detection bit vector to the bounceback detection  
9 bit vector associated with the eliminated domain member.

1 24. The method of claim 21, wherein the step of setting the bounceback  
2 detection bit vector of an eliminated domain member to indicate which variable caused  
3 that domain member's elimination includes:

4 based on the constraints, identifying a join corresponding to a conjunction;  
5 logically ORing the bounceback detection bit vectors associated with the domain  
6 members included in the join thereby producing a resulting bounceback  
7 detection bit vector; and  
8 copying the resulting bounceback detection bit vector to the bounceback detection  
9 bit vector associated with the eliminated domain member.

1 *Def 017* 25. The method of claim 21, wherein the method is implemented by a set of  
2 software instructions running on a computer

*add  
B2*